1985

# Local area network architectures for distributed processing /

Dennis C. Ebersole
*Lehigh University*

# LOCAL AREA NETWORK ARCHITECTURES FOR DISTRIBUTED PROCESSING

by

Dennis C. Ebersole

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computer Science and Electrical Engineering

Lehigh University

1985

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

May 15, 1985
DATE

Kenneth K. Tang
PROFESSOR IN CHARGE

Eric L. Thompson
CHAIRMAN OF DEPARTMENT

ACKNOWLEDGEMENTS

The guidance of Professor Kenneth K. Tzeng, supervisor of this thesis, is gratefully acknowledged. His assistance in finding sources, providing a focus for the thesis and suggesting changes were invaluable to the author.

Northampton County Area Community College is also gratefully acknowledged for providing educational assistance and granting a sabbatical leave to the author to pursue his graduate studies.

TABLE OF CONTENTS

ABSTRACT

This thesis is confined to the consideration of local area networks (LANs) to implement distributed processing. The International Standards Organization's Open Systems Interconnection Reference Model, a seven-layer architecture for computer networks, is utilized to structure the discussion. For the two lowest layers of the architecture the IEEE Project 802 Committee Standards for LANs provide the framework for the presentation.

An overview which defines the term distributed processing and provides numerous applications showing the usefulness of distributed processing is presented. The current status in the development of standards for each layer is given. The functions of each layer are defined apropos a local network environment. Possible implementations of each function are presented. The interfaces between layers are defined and illustrations are given. Predictions are made about changes that can be expected in the near future with the advent of various technological advances.

The advantages of using a layered architecture in the development of a LAN for distributed processing are shown. The need for standards accepted by the industry is demonstrated by illustrating how difficult the interconnection of networks is when standardization is not present.

# 1. INTRODUCTION

When computers are connected by a communications network and integrated into a system by a network-wide operating system, distributed processing can occur. Distributed processing has numerous advantages. Expensive hardware and software can be shared, which reduces overall system costs. Applications include electronic mail, distributed databases, specialized workstations, parallel processing, distributed computation, process control, factory automation and project planning. A distributed system provides greater reliability and extensibility through system redundancy, rapid reconfiguration, and crash recovery capabilities.

Although distributed processing is possible with a widely dispersed network this thesis will deal almost exclusively with issues relevant to a local area network (LAN) because long-haul networks have been used for many years and are fairly well-defined and researched and because many advantages to be gained by distributed processing only apply to LANs. A local area network is a communications network confined to a small geographic area such as an office building, production facility or campus that is owned and operated by a single organization. It is not the interconnection of devices on a desktop or components within a single piece of equipment. LANs use a physical communications channel with a high data rate and a low error rate when compared with telephone lines used for data communication. [3] LANs can be configured using various topologies such as a star (where all

communications are forwarded through the central computer), multidrop line, ring, hierarchical (tree-like structure), or mesh (multiple paths between computers on the network.) [4]

Several industry analysts are predicting that LAN sales in 1985 will approach three times the 200,000 sold in 1984, so it is imperative that industry standards be finalized shortly. The standards developed or being developed follow the International Standards Organization's Open Systems Interconnection Reference Model which specifies a seven-layer architecture for networks.[1] The IEEE Project 802 Committee has been working on LAN standards that encompass the two lowest layers of the reference model. By using layers in the development of a LAN, as new technologies become available it will be possible to change certain aspects of lower layers with no impact on higher layers. For example, applications software written for a LAN using coaxial cable for its transmission medium would not have to be changed if the transmission medium was changed to optic fiber. Also, if LANs are built using the standards, relatively few gateways and bridges need to be developed to interconnect different LANs and each will have a built-in market for the product.

For pedagogical reasons, a top-down approach is chosen in discussing distributed processing in the remainder of this thesis. By discussing

---

[1] An open system is one built to a published specification, while a closed system is completely proprietary. [5]

the application layer first, a better idea of what a distributed computer system can and should do and also why the presentation layer must perform the functions it does is possible. Similar reasoning applies to the interface between lower layers.

Chapter 2 provides an overview of distributed processing. The advantages of distributed processing over centralized will be presented. An overview of the Open Systems Interconnection Reference Model and IEEE Project 802 Committee standards is presented, along with predictions on future trends in distributed processing. The functions of the application layer are presented in Chapter 3. Presentation layer functions are described in Chapter 4. Chapter 5 explains why the session layer is relatively unimportant in a local area network. The functions of the transport layer will be outlined in Chapter 6. Chapter 7 describes those functions of the network layer that are important in a local area network. Chapter 8 describes the data link layer standards defined by the Project 802 Committee for three meduim access control methods. The physical layer for these three local area networks is presented in Chapter 9.

## 2. OVERVIEW

Distributed processing refers to a computing environment in which a collection of processor - memory pairs are connected by a communications network and logically integrated by a distributed operating system to implement a given application. [1, 2] The rationale for considering distributed processing in this thesis is the potential advantages inherent in distributed processing.

### 2.1 Advantages

With VLSI (very large scale integration) technology very powerful, inexpensive microcomputers are becoming available. Thirty-two bit microprocessors are already on the market and a mass-produced 1-megabit chip should be available in 1986 - maybe sooner. Every executive/manager will be able to have his own personal computer, but will want access to data stored on other computers. By networking these computers together and setting up a distributed computer system expensive resources can be shared. Expensive peripherals such as high speed letter quality printers and hard disks can be shared. Systems and applications software such as compilers, spreadsheets and word processors can be shared. Databases can be distributed and data and files can be gathered or created at the most efficient location and transferred to the appropriate individual in the relevant format as needed. Each computer's operating system can be simpler [1] and thus more reliable, the operating system does not have to be general purpose but can be designed for specialized applications. The distributed

operating system can decide which processor will be most efficient for each task based on processor designs and current loads. [2] Performance is also enhanced by performing tasks in parallel. [1] By decomposing computations and assigning various components (transactions) to different processors based on a heuristic scheduling algorithm throughput and response time can be improved dramatically. [2]

A distributed system also provides greater reliability and extensibility. Reliability is enhanced by system redundancy. If one computer on the network fails, the network can still function. Another computer which shares memory with the failed unit can assume its responsibilities. Data and software can be stored in several distributed locations to allow for easy recovery from an isolated crash.

Extensibility refers to the ability to add to a system easily and inexpensively. Most distributed computer systems can be reconfigured as part of the normal operation of the network with no noticeable degradation in system response and no need to alter the distributed operating system.

Finally, a distributed computer system is most appropriate for certain applications. Electronic mail has proven to be more efficient than interoffice mail for such communications. Many real-time processing applications require distribution by their nature. Process control requires sensors and actuators in the plant connected to a computer.

By having one microcomputer for each location as part of a distributed computer system to monitor the readings and make needed adjustments based on all available readings, problems encountered when a single computer is used such as the need to multiplex inputs and outputs can be alleviated. [1]

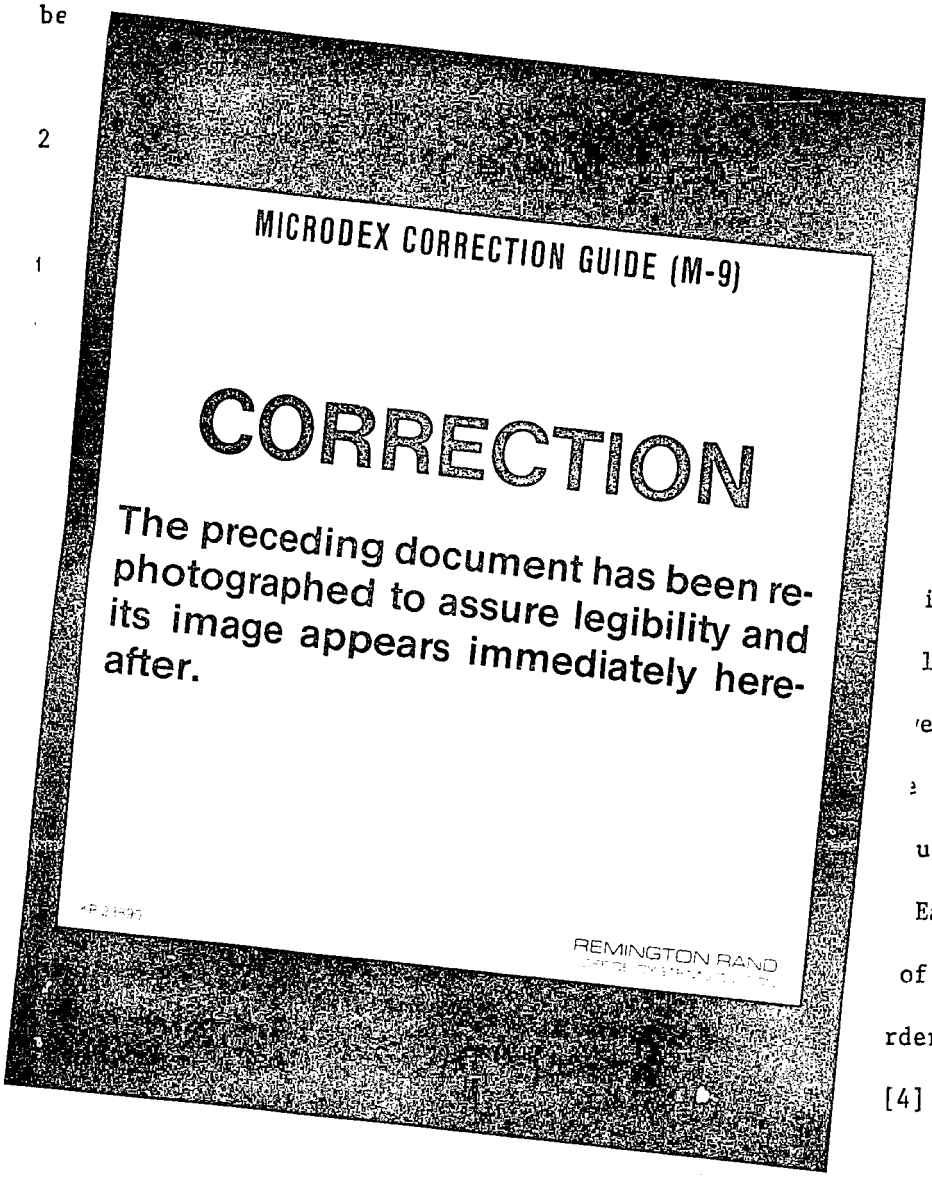
## 2.2 Open Systems Interconnection Reference Model

If two networks are built to a standard specification, it is easy to interconnect the two networks to allow for even greater communication. Since most LANs currently marketed are proprietary it is difficult to interconnect them; special software must be developed to allow users of different networks to communicate.

The term network architecture refers to the "precise definition of the functions that a computer network and its components should perform...specified in terms of protocols for communication between pairs of peer-level layers" [4] or peer processes and the interface between adjacent levels of the structure. The rules and conventions used by peer processes when communicating is called a protocol. [6] Each protocol consists of syntax (frame formats), semantics (set of requests, actions and responses possible) and timing (the order in which events can occur during the communications process). [4]

OSI is a seven-layer architecture. The diagram on the following page illustrates the layers, protocols and interfaces of the architecture. [8]

By having one microcomputer for each location as part of a distributed computer system to monitor the readings and make needed adjustments bas[illegible] available readings, problems encountered when a single cor[illegible]'s can be




2

1 is easy
unica-
s diffi-
to allow

inition of
ld per-
'een pairs
e between
used by
Each
of
rder in
[4]

e following
page illustrates the layer[illegible]. he architec-
ture. [8]

By having one microcomputer for each location as part of a distributed computer system to monitor the readings and make needed adjustments based on all available readings, problems encountered when a single computer is used such as the need to multiplex inputs and outputs can be alleviated. [1]

## 2.2 Open Systems Interconnection Reference Model

If two networks are built to a standard specification, it is easy to interconnect the two networks to allow for even greater communication. Since most LANs currently marketed are proprietary it is difficult to interconnect them; special software must be developed to allow users of different networks to communicate.

The term network architecture refers to the "precise definition of the functions that a computer network and its components should perform...specified in terms of protocols for communication between pairs of peer-level layers" [4] or peer processes and the interface between adjacent levels of the structure. The rules and conventions used by peer processes when communicating is called a protocol. [6] Each protocol consists of syntax (frame formats), semantics (set of requests, actions and responses possible) and timing (the order in which events can occur during the communications process). [4]

OSI is a seven-layer architecture. The diagram on the following page illustrates the layers, protocols and interfaces of the architecture. [8]

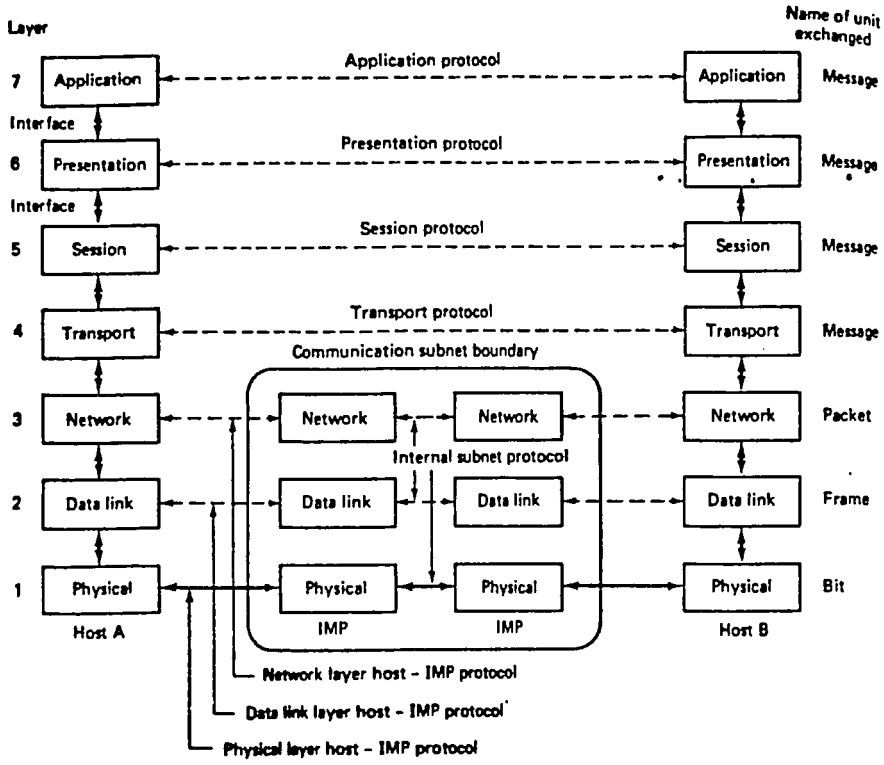



FIGURE 1

To understand how peer processes "communicate", consider the following scenario. John, the president of Company A, decides to set up a meeting on a proposed merger with Joan, the president of Company B. He jots down a proposed agenda and gives it to his assistant. The assistant checks John's calendar and adds a proposed place, date and time to the note. The assistant gives the revised message to the president's secretary who calls Joan's secretary at Company B. Joan's secretary copies the message and gives it to Joan's assistant. He notes the proposed place, date and time and forwards the agenda to Joan. Joan suggests minor changes to the agenda and gives the message to her assistant. Her assistant checks the proposed date and time and

adds to the message that the date and time are okay. The revised message is given to Joan's secretary, who calls John's secretary to relay the message. John's secretary gives the note to John's assistant who marks John's calendar and gives Joan's suggestions to John.

In this example there are three layers of communication. John and Joan communicate about the agenda. John's assistant and Joan's assistant communicate about the place and time of the meeting. The two secretaries relay the messages be telephone. The conventions used by peers in each company when communicating are the protocols. The communication within each company between "layers" is the interface between the layers. Note that only the two secretaries actually converse with each other. This corresponds to the physical layer and the telephone line is the communications medium. The communication between the two assistants and the communication between the two presidents is virtual communication.

Each layer in OSI is independent of the other layers. If a lower layer changes the method of performing a certain service, higher levels won't have to be altered; they only expect a certain level of service. This allows engineers and programmers to concentrate on one layer at a time in designing a network much as is done in structured, top-down programming. Each layer owns a header in each packet to be sent which contains the information needed to communicate with its peer process. [4] A header is information added to the data being sent for control purposes.

## 2.3 IEEE Project 802 Committee Standards

While the OSI Reference Model assumes the presence of a subnet which must be traversed in order for the actual bits of information to get from one user to another, which is the usual case with long-haul networks, IEEE's Project 802 standards deal with local area networks where one user communicates more or less directly with another user. In a local area network there is no need to have a subnet protocol and a host-subnet protocol. Every user on the LAN uses the same network protocol.

Project 802 defines sublayers for three types of LANs commonly used. These three sublayers encompass the first two layers of the OSI Reference Model as shown below.

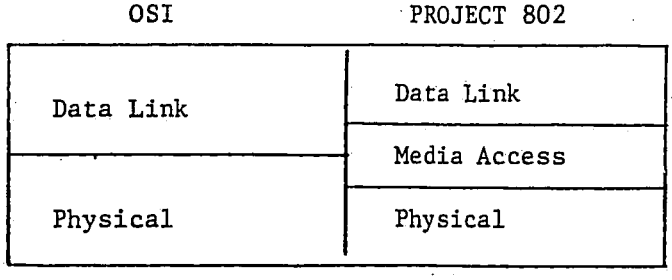



FIGURE 2

Since higher layers are not specified by this standard, it is still possible for two networks to be incompatible even though they meet the standards. In addition, the standard defines three incompatible LANs, a passive bus very similar to Xerox's Ethernet, a bus which uses token passing and a ring using token passing. [7] (Each of these

LANs will be explained in detail later.) In addition, two more LANs have been proposed for inclusion in the standard and are receiving serious consideration - a star network with a gateway to the ethernet-like LAN and a cheaper version of the ethernet-like LAN which is incompatable with the ethernet-like LAN. [7] When you add to this the fact that nearly 60% of the LAN's currently in operation use a twisted pair medium with an ethernet-like access (Corvus' Omninet) [7] and you will understand why so much work is still ongoing apropos networking standards. A step in the right direction was the announcement by Xerox, DEC (Digital Electronics Corporation) and Intel that their local area network products would all conform to IEEE's ethernet-like standards and thus be compatible. [7] IBM's announced entrance into the LAN market may also help stabilize the marketplace by establishing a defacto standard.

Using the OSI Reference Model for the upper layers of the architecture does not solve the problem of incompatible distributed computer systems either. Proposals for the presentation layer are only now being considered and may not be adopted for a year or more. The application layer protocols are far from being finalized. Until these layers are standardized you can expect that most commercial LANs will only allow a limited number of types of computers on the network and internetworking LANs from different manufacturers will require a significant programming effort on a case by case basis - a patchwork job rather than a universal solution.

## 2.4 Future Trends

A number of trends will impact the direction that distributed processing takes. More and more of the network architecture is being implemented in ROM chips. This makes networking less expensive but also means that minor revisions cannot be easily made in the network software; it is firmware, not software. In addition, some microcomputer manufacturers have started to include networking software with the base computer. Again this tends to reduce the cost of networking, but unless there is an industry standard the purchaser may be compelled to use a LAN supported by the software included with the microcomputer.

The increase in LANs has alarmed the software industry. What is to stop an organization from purchasing one copy of an expensive applications package, putting it on their LAN and allowing multiple users to use the package simultaneously. A number of solutions have been proposed or tried. Some companies offer network licenses for their software that authorize an organization to use the software on a network at a substantial savings when compared with the cost of multiple copies for individual users. Other manufacturers are working with computer and network companies to limit the number of copies of a software package that can be running simultaneously. Or the software could be encrypted and the end user could be charged for the key needed to use the software. Finally, optic fiber technology may radically alter networking because of its high bandwidth. Standards for networks using such high bandwidth are currently being considered.

# 3. THE APPLICATION LAYER

The application layer, as the name implies, is dependent upon the needs of two users or user programs. A standard for this layer should define those functions of a general nature that could be incorporated in a network library and chosen as needed by two applications. Some issues addressed at this level include network transparency, distributed computation, distributed communication, heterogeneity, distributed database management, the network operating system, a distributed operating system, distributed real-time systems and crash recovery. In other words, this layer provides application-oriented and management functions needed to initiate, maintain, and terminate communication between two application processes. [4]

## 3.1 Network Transparency

Network transparency refers to the hiding of the physical distribution of network resources from the user. [8] As far as the user is concerned all resources needed should appear to be resident resources, even if the data is stored on several disks, in different formats, the program(s) on another disk and several processors were used in doing the calculations. The brand names of the peripherals and computers is of no concern to the users in a transparent environment.

## 3.2 Distributed Computation

Distributed computation refers to the decomposition of large scale problems into smaller problems which can be assigned to various proces-

sors allowing for multiprocessing. [2] There are two ways to distribute the computation to be done. In a sequential trace the operations are done in the "same sequence" that they would be done on a single processor but on multiple processors as indicated below where operations 1, 2 and 3 are done before 4 and 5.

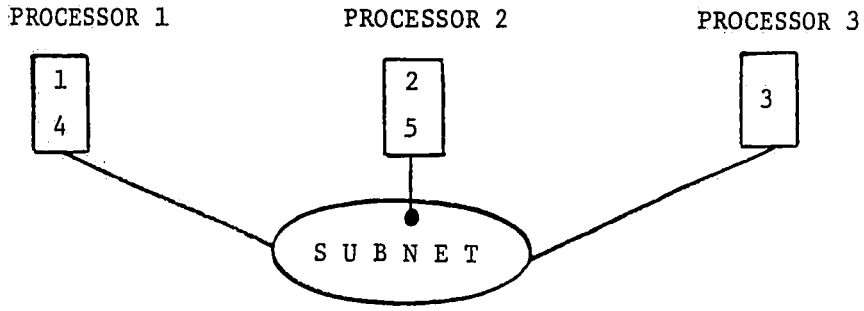



FIGURE 3

A second method clusters processes using some algorithm to determine the best placement and has the processes communicate with each other as needed to complete the computation correctly. Processes on different processors need only wait if an input is needed from another process as indicated in the example below. [2]

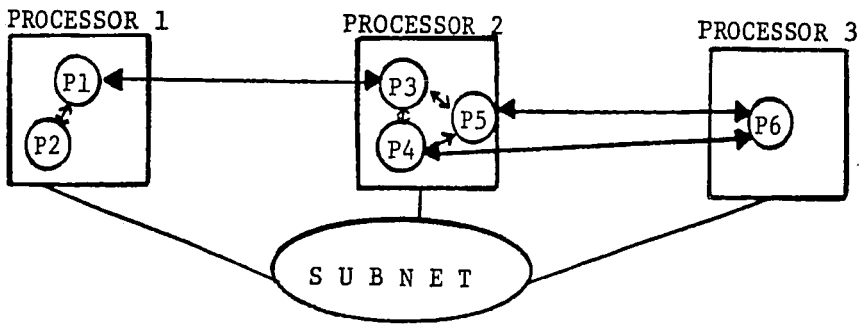



FIGURE 4

This requires concurrency control protocols and the need to group actions into transactions, a sequence of actions which cannot be run on two different processors concurrently because they must be performed sequentially or because they need access to a group of resources prior to execution. This is necessary because otherwise the system might experience a deadlock. A deadlock is a situation in which no progress can be made and it is impossible to get out of the situation without manual intervention. For example, suppose process 1 needs resources A and B before it can proceed and it has acquired resource A and is waiting for resource B to be freed before proceeding. Also process 1 will not release A until it has executed to completion. Meanwhile process 2, which needs resources A, B and C has acquired resources B and C and is waiting for A to become free. Since process 1 won't release A and process 2 won't release B, a deadlock has occurred. On the other hand it may be possible and desirable to allow two processes to access the same resource simultaneously. For example, reading data on a disk is nondestructive and simultaneous reading should be supported.

### 3.3 Resource Sharing

Techniques that can be used in resource sharing include lock tables, time stamping requests and validation. [2] With lock tables, a process marks a resource as locked when it has been acquired so no other process can access the resource. With time stamping, a request is stamped with a unique order number - for example the current time plus the unique process number - and requests are granted in order

based on the time stamp. With validation through a series of communications a process's right to use a resource is verified.

Another concern at this level when resources are shared is how to limit access to a resource such as sensitive data. One method that can be used is the login approach with an access control list or matrix. During login the user gives a login name, password and/or account number. This information is then used along with the access control list to determine if the user is allowed access to the resources desired. Even communication on certain paths can be denied. [9] The access control list can reside at one or more network access monitors or each computer on the network can have a copy. Only the network operating system should be allowed to alter the list. [9] Some disadvantages of this approach is that each user must have an account on each system he wants to use prior to using the system (destroying some of the network transparency) and the potential for an extremely large list that has to be updated frequently.

Volumes, files or even records can be marked as public, read-only, private, group-owned, etc. If private or group-owned a password or user authentication can be required prior to allowing use of the information. This information is again stored in an access control list.

Another approach is to use capabilities (permission slips). If a user process has been given the requisite capability it can access the

resource. The network operating system controls the issuance of capabilities.

A final approach that can be used is to assign security levels to resources and security clearance levels to processes. This sets up a hierarchical structure which may work well for some organizations. Each user has a security clearance and the user's processes inherit this clearance. A user with top security clearance (e.g., the president of the organization) can access all resources whereas a user with low level clearance is limited in the resources he can access. To maintain security copied files, etc. inherit the security level of the source file, etc.

### 3.4 Distributed Communication

Distributed communication is needed when distributed controllers must cooperate. Some examples of such applications are the start-up of a distributed operating system, crash recovery techniques, scheduling of processes, distributed access control and distributed routing and congestion control. [2]

In each example above each computer on the network would have a copy of a sophisticated algorithm and would use control information coming from the other computers on the network to decide what to do.

### 3.5 Heterogeneity

Heterogeneity refers to incompatibilities of equipment on the network. Ideally it should not matter what computers are attached to

the network, how the files are organized or what access rules are used at each host. The network should convert one machine language to another, one character set to another (e.g. ASCII to EBCDIC), one file organization to another file organization needed by an applications package, etc. in a transparent fashion. To allow for heterogeneity, a standard data format is often used on the network. Each station on the network must perform the necessary conversions to put data onto the network and take data off the network to be used by the processes at the station. A second approach is to have an intermediate translator which accepts data from the source station and converts it into an acceptable format for the destination. [2]

A disadvantage of the second approach is the need to have two translators for every pair of incompatible stations. If these translators have to be located at each station a high overhead can result when the mix of stations is very heterogeneous. Also, with this approach, adding a station which is incompatible with every station currently on the network requires implementing a new translator at each station. With the first approach the new station could be added immediately if it has a translator to put data on the network and one to take data off the network.

## 3.6 Distributed Databases

A distributed database is a database that is geographically distributed but logically integrated. [2] A distributed database system refers to the logical integration of geographically distributed

databases. Of course, each database in the system may be distributed. In some cases the physical distribution occurs naturally. Each area or department of an organization may have had their own computing facilities originally but the need to access data stored in other areas' computers and/or the desire to eliminate data redundancy necessitates networking the computers to allow access to and coordination of the various databases. Since most accesses will be to the "local" database and updates of the database usually occur locally, an efficient system is relatively easy to design.

Some issues that must be considered in designing the system include access control, redundancy, concurrency control, maintenance of database integrity, and transparency. As discussed previously, access control is a concern in any distributed system. With databases there is the additional problem of updating the database. Should non-local users be allowed to update a database? If certain users may create, delete or change files or records, how does the distributed database system know which user is authorized to perform which operation on which database? Or are updating functions limited to the database manager. The choice may depend on the issue of redundancy. Anyone who has used a computer system for more than a week knows the advisability of backups. If certain data must be available on a database at all times, that data should be stored at several (distributed) locations. Then if one station which houses this data crashes, the data can still be accessed at the second station. This is called data redundancy. The system may also incorporate equipment redundancy, so if one piece

of equipment fails it can be replaced by its backup with little or no degradation in system response. (This approach is utilized by NASA on space flights where three identical computers are used.)

The goal of concurrency control algorithms is to allow the maximum amount of simultaneous access/processing of a distributed database possible without introducing errors into the database. Clearly, if more concurrency is permitted, the performance of the database will improve since more transactions can be handled in a specified period. However, unlimited concurrency can lead to problems. For example, suppose an airline has one seat left on a flight to Los Angeles. If two travel agents access the airline database simultaneously, they will both read that one seat is available. Both agents might then reserve this seat for their client. The agent who submits his or her reservation last will overwrite the reservation made by the other agent. There will be two tickets issued for the same seat - an intolerable situation. Various techniques have been proposed for controlling concurrency. They include locking a portion of the database, use of a token, polling, reservation schemes and limited contention protocols. [2] Some general concerns of all such algorithms are granularity and how distributed the algorithm is. Granularity refers to the size or amount of the database that can be protected by a single locking or reservation. If the entire database is protected, no concurrency is possible. If each tuple in a relational database can be individually protected, a large amount of concurrency is possible but the algorithm must be very complex because a process may have to acquire many tuples

before proceeding, so deadlocks are possible and must be avoided. Similarly, a centralized control algorithm which facilitates protecting portions of the database from concurrent access using a single controller is relatively easy to implement but some of the advantages of a distributed database are lost in the process. Distributed algorithms are more complex but retain the distributed database attributes.

Database integrity refers to the correctness of the database. A well-designed concurrency control algorithm insures the correctness of the database during concurrent processing. But how can database integrity be maintained if a crash occurs while the database is being updated? One approach is to classify certain actions as atomic. Atomic actions must be completed or not done at all. [8]

Prior to performing an atomic action a checkpoint is set up to save the current state of the database region affected by the atomic action. [9] If a crash occurs in the middle of an atomic action, the database is restored to its status at the checkpoint. In a distributed environment all transactions that occur from the time of the crash until the crashed station comes back on-line are saved and sent to the station when it returns so it is up-to-date. [8]

A distributed database is transparent if users of the database do not have to know the physical characteristics of the database. In fact, as far as the user is concerned it should appear that the entire

database is located on his machine. The network should hide from the user the fact that the database is spread over a relatively large geographic area as well as other implementation decisions. There should be no difference in user interface with the database between a centralized database and a distributed database.

## 3.7 Network Operating System

When designing a distributed system there are essentially two choices apropos the operating system - a network operating system or a distributed operating system. A network operating system consists of software added on top of the computer's existing operating system. A file server must be written for each different operating system on the local area network. Every request is checked by the file server and sent either to the local operating system or onto the network to be processed depending on what was requested. [5] A network operating system has the advantage that each station on the network can still run application programs for the resident operating system without change. It is also relatively easy to implement. [8]

## 3.8 Distributed Operating System

A distributed operating system, on the other hand, is a single native operating system for all the distributed stations. [2] There are two models commonly used for distributed operating systems - the process model and the object model. In the process model program processes communicate with server processes that manage network resources such as files, printers and disks. Each server process has a

unique name. An algorithm is used to determine which program process is served if several processes request the same resource. In the object model all network resources are considered objects. Each object "has a type, a representation, and a set of operations that can be performed on it." [8] In this model the distributed operating system manages permission slips (called capabilities) which allows the processor to perform a certain operation on an object. By having only one capability for operations such as writing on a disk file where synchronization is important and multiple capabilities for operations such as reading which need not be synchronized, system integrity can be maintained while allowing for concurrency where possible to improve system performance. Such a system must ensure that a malicious user or malfunctioning station cannot generate capabilities at will or the system will fail.

# 4. THE PRESENTATION LAYER

The presentation layer is an end-to-end layer which provides useful but non-essential services to network users. Examples of services provided by this layer are text compression, encryption, terminal conversion, file and record transfer management, user authentication and character code and file conversion. Although this layer has not been standardized, proposed standards have recently been sent to the International Standards Organization, so standards for this layer may be published shortly. Services available at this level are usually implemented in system library routines which can be activated by users as desired.

## 4.1 Text Compression

Although text compression is not currently a concern on LAN's because of the bandwidth available to users, as more users are added to existing LAN's bandwidth will become a more valuable resource and text compression will become desirable. In addition, text compression can be useful for data security and may be used even if efficient use of the bandwidth is not a consideration. Three examples of text compression are Huffman Coding, context dependent encoding and run length encoding.

Huffman Coding utilizes an algorithm designed to have symbols with a high probability of occurring assigned short codes and low probability symbols assigned long codes. This will result in sending fewer

bits per message on the average. The algorithm sets up an n-ary tree with the (finite set of) symbols to be encoded placed at the leaves of the tree. A symbol's location on the tree determines its coding. To illustrate the algorithm, suppose you have nine symbols you wish to encode with occurrence probabilities as shown in the table below. Also assume that a binary tree is to be used.

TABLE 1

| SYMBOL # | PROBABILITY | HUFFMAN CODE |
|---|---|---|
| 1 | .25 | 00 |
| 2 | .25 | 10 |
| 3 | .15 | 010 |
| 4 | .1 | 110 |
| 5 | .08 | 0110 |
| 6 | .07 | 0111 |
| 7 | .05 | 1110 |
| 8 | .03 | 11110 |
| 9 | .02 | 11111 |

To start the process the two symbols with the lowest probability (8 and 9 above) are assigned to nodes and connected to a new node by two arcs. The probability of the new node is the sum of the probabilities of the two symbols (.03 + .02 = .05). The tree so far is shown in figure 5.

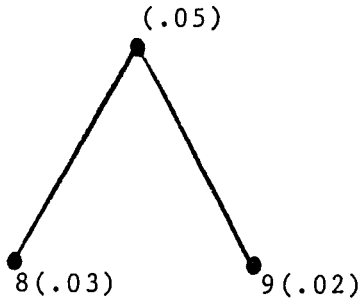



FIGURE 5

As symbols are assigned to a node they are removed from further consideration and the new mode which has a probability equal to their sum replaces them. The two lowest probabilities in this new set are again connected to a new node with probability equal to the sum of the probabilities of the two connected nodes. This is repeated until every symbol is assigned to a leaf of the tree. In our example it results in the following tree.

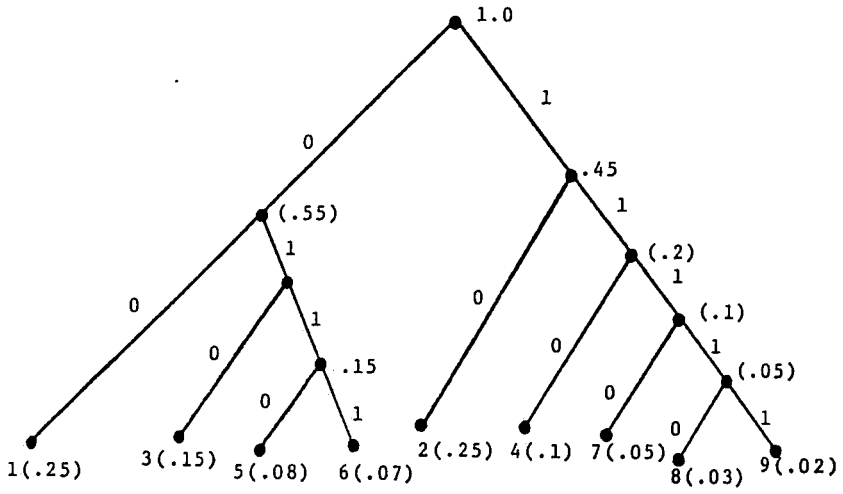



FIGURE 6

If codes are assigned to the leaves of the tree by using the convention that each left branch taken is coded as a 0 and each right branch as a 1, the codes in Table 1 result. The average number of bits used is $(.25)(2) + (.25)(2) + (.15)(3) + (.1)(3) + (.08)(4) + (.07)(4) + (.05)(4) + (.03)(5) + (.02)(5) = 2.8$. If no compression was used, 4 bits would be needed for each symbol - a savings of 30%.

Context dependent coding utilizes the fact that the probability of encountering a certain symbol in many applications is dependent upon the preceding symbol. (Huffman Coding assumes the probabilities are independent.)

By dividing the symbols into groups such that if the current symbol is in a certain group then there is a high probability that the next symbol is also in that group, the number of bits needed for each symbol is reduced and the average number of bits sent per symbol is reduced. This is because the same string of bits will represent a different symbol in each group. Special symbols are used to switch from one group to another. For example, suppose 100 symbols can be grouped into four such groups of approximately equal size. Then six bits can be used to encode the symbols in each group. If the bit patterns to change groups are also six bits long and a group change occurs on the average after six symbols are sent, an average of 42 bits are needed to send 6 symbols or an average of 7 bits per symbol. Since 8 bits are needed if no compression is used, this is a savings of 12.5%.

Run length encoding utilizes the fact that certain symbols may be repeated in succession fairly often. Examples are 0 bits in binary files and blanks in text files. By encoding the number of such symbols rather than the symbols themselves, significant savings can result. Andrew Tanenbaum gives the following example in his book, Computer Networks which results in a savings of 34%. The bit string:

0001000001000000100000000000000100000100010000000110100000101

would be encoded as

011 101 110 111 111 000 110 011 111 000 000 001 101 001

using 3-bit symbols. Each 3-bit symbol except 111 indicates the number of 0 bits before the next 1 bit. The 3-bit sequences, 111 must be added with the next 3-bit sequence(s) to get the number of 0 bits before the next 1.

## 4.2 Data Encryption Techniques - Single Key

If sensitive data is being sent on a LAN, some form of security is needed. Data encryption transforms the original data in order to make it difficult for someone without the encryption key to determine the original message. Hardware devices are often used at the data link layer to encrypt and decrypt messages sent on a network. If this encryption is not available or not secure enough, the user may request end-to-end encryption. Some well-known encryption techniques are the Data Encryption Standard (DES) and public key encryption.

To understand DES it is first necessary to know a little about cyphers. A transposition cypher rearranges the characters in the

original text (called plaintext) to form the encrypted text (called ciphertext). [10] For example, the plaintext "The dog barked" might become the ciphertext "dTgehd b eokar", if the order of output was 5, 1, 7, 13, 2, 14, 4, 9, 8, 3, 6, 12, 10, 11; i.e., the fifth input character is sent first, then the first, then the seventh, etc. By knowing this reordering (called the key) the receiver can easily decipher the message.

In a substitution cipher one character is replaced by another. [10] For example, every "a" might be replaced by a "q", every "7" by a "p", etc. To decipher the message the receiver must know the key - which is to replace every "q" by an "a", every "p" by a "7", etc. A polyalphabetic substitution cipher uses multiple substitution alphabets. [10] For example, if five substitution alphabets are used, the first, sixth, and eleventh character, etc. would use the first alphabet to determine the substitution, the second, seventh, and twelfth character, etc. would use the second alphabet and so on. A digraphic substitution cipher replaces pairs of characters by their substitution pair. [10]

A product cipher uses a combination of substitutions and transpositions. A block cipher divides the plaintext into fixed size blocks to be encrypted. The same block will always result in the same encryption. [10] With a stream cipher both the source and the destination station operate in the encryption mode. By maintaining a shift register at the destination station identical to one at the

sender and using the encrypted message from the previous character in conjunction with the current character, the plaintext can be determined. In this manner the character sent is dependent upon the previous character, so the same block of plaintext will result in different ciphertext because the encryption depends on the state of the shift register. [8] In a synchronous stream cipher, the ciphertext is dependent upon a character's position in the stream as well as the character. Sender and receiver must remain synchronized to avoid losing data. [10]

The Data Encryption Standard (DES) is a block cipher that operates on 64-bit blocks. DES uses a 64-bit key of which only 56 are used in the encryption process. There are thus $2^{56}$ different keys possible. The encryption process involves a permutation on the 64-bit plaintext followed by 16 separate product ciphers and then the inverse of the initial permutation. The process is reversed to decrypt the message. [10] DES is usually implemented in hardware at the data link layer and therefore encrypts all bits that are sent - data, headers and trailers. Many researchers feel that DES is not secure enough for many applications, which is why end-to-end encryption is often recommended. [8]

## 4.3 Data Encryption Techniques - Public Key

One of the problems with single key algorithms such as mentioned above is how to distribute the key while maintaining the necessary secrecy. On a LAN, two users at two stations may want to communicate

privately for several seconds. It is clearly undesirable to require that a key be physically delivered by the initiator of the conversation to the other user in order to allow for secure communication. An ingenious solution to the problem is to use a two key algorithm called public key encryption. Each user publishes a public key (which can be changed periodically) to be used in communicating with the user. For example, if user A wishes to communicate with user B, he encrypts his message using user B's public key. Upon receiving the message user B decrypts the message using the second, private key which is the inverse of the public key. The security of the algorithm hinges on the difficulty in determining the private key from the public key.

In 1978 some researchers at MIT published the RSA algorithm which can be used to generate the two keys needed for public key encryption. Their technique relies on the difficulty in factoring large numbers. For example, it took a Cray-1 supercomputer over 32 hours to factor a 69-digit number into primes. [11] The following example will illustrate how the algorithm works, although in actual practice each prime chosen would have at least 100 digits. [11]

| | |
|---|---|
| 1. Pick two primes. | 47 and 251 |
| 2. Find their product. | $47 \cdot 251 = 11{,}797$ |
| 3. Subtract 1 from each prime and find the product, z, of these two numbers. | $z = 46 \cdot 250 = 11{,}500$ |
| 4. Find a number, s, which has no whole numbers in common with this number except 1. (Such numbers are said to be relatively prime.) | s = 187 is such a number. ($187 = 11 \cdot 17$, while $11{,}500 = 2^2 \cdot 5^3 \times 23$) |

| | | |
|---|---|---|
| 5. | Find another number, p, such that p and s are inverses modula z; i.e., $p \cdot s = 1 \bmod z$. | p = 123 works, since $123 \cdot 187 = 23{,}001$ and $23{,}001 \bmod 11{,}500 = 1$ |
| 6. | The public key is the pair (p,n) and the secret key is s. | (123, 11797)<br>187 |
| 7. | To send a message, compute $C = M^p \bmod n$ for each character, M, in the message. This is the ciphertext. | If M = "A"( 64 in ASCII), compute $64^{123} \bmod 11797 = 5362$. 5362 is the ciphertext. |
| 8. | When the encrypted message is received, compute $M = C^s \bmod n$. (Recall that p and s are inverses.) | $5362^{187} \bmod 11797 =$ 64 ( "A") |

An additional feature of public key encryption is that it can be used to authenticate users. If all users are required to use their private key to encrypt messages, then the fact that such a message can be decrypted using the public key guarantees that the sender was who he said he was (assuming the private key has remained secret). By using the receiver's public key to encrypt the message a second time, message security can still be maintained.

## 4.4 Virtual Terminal Protocols

Since there are over 100 different terminals being sold and new terminals are marketed with technological advances, it is important for a LAN to support different terminals. This requires a Virtual Terminal Protocol (VTP). ISO is currently working on a standard in this area. Such a protocol must support scroll mode, page mode and data entry terminals. [12] To do this most protocols use the concept of a network virtual terminal and application programs are written to assume that their terminal is a network virtual terminal. When two users with

different terminals wish to communicate, the terminals must negotiate which options will be in effect. In essence they must determine a lowest common denominator of terminal services that both can support. Default values can be used to speed up the negotiations; if neither side picks a non-default option, the default option is used. In addition, the results of a previous negotiation can be saved to eliminate the need for future negotiation as long as neither terminal changes. The VTP should allow new terminals to be added to the LAN without major changes. [8]

## 4.5 File Transfer Protocol

Just as communication between different stations requires a Virtual Terminal Protocol to hide the differences in equipment, a File Transfer Protocol (FTP) is needed to hide differences in file systems on the LAN from users who wish to transfer distant files to their station for their use. The FTP should give the user the illusion that there is a single file system on the LAN by handling various size files, various datatypes and structures, etc. [2] It should support typical commands such as copy to , copy from, append to, rename, delete, list directory and abort as if the files were local even when they reside at a distant station. The protocol must check that the user is authorized to access the file and should allow partial transfers. The option of specifying a foreground or background transfer may be desirable. To do this there are typically three phases to the transfer - negotiation of options to be in effect, authentication of the user, and the actual transfer. [9] Some examples

of FTP's are the ARPANET FTP, NSA FTP and NSW File Package Protocol. [9] Although ISO is working on a standard for file transfer, there is no current standard and the current protocols are not compatible. [13]

## 4.6 The Mainframe - Microcomputer Link

A final problem at this layer yet to be resolved is the mainframe - microcomputer link. The mainframe world uses the EBCDIC character set which has 256 different characters, while microcomputers use ASCII which has 128 different characters. Clearly, text files on a mainframe cannot be downloaded (copied from a mainframe to a microcomputer) directly. Characters common to both sets must be converted from EBCDIC representation to ASCII representation. Also, a decision must be made on how (or if) to represent EBCDIC characters which have no counterpart in ASCII. In addition, file or database formats used on the mainframe may be incompatible with the format expected by a proprietary applications program such as a spreadsheet on the microcomputer. Although programs have been written to make the conversions necessary when downloading files to be used by several popular applications program, no general solution to this problem is presently available.

## 5. THE SESSION LAYER

The session layer establishes an end-to-end connection between two entities, manages the dialog between them including synchronizing data exchange and operations, authenticating processes and recovering from broken connections, and disconnects at the end of the session. [4] This layer also handles the buffering and segmenting of large transmissions. [14] A session layer header is added to user data. This header contains the source and destination addresses, a sequence number for segmented transmissions, billing information and a description of the data. [10] By buffering messages sent and using the sequence number this layer can set up a new transport connection when the old one fails and only retransmit the lost segment(s). Since such segments may arrive out of order, the session layer is responsible for ordering the segments before sending them up to the presentation layer. Even though only a minimal session layer standard has been approved by the ISO [4], some of the tasks of this layer have already been implemented in VLSI chips. [5] When a complete standard becomes widely accepted we can expect to see many manufacturers develop such chips which will further reduce the cost of LANs.

# 6. THE TRANSPORT LAYER

The transport layer provides transparent end-to-end control of message exchanges between users [10], assembles and disassembles packets to be sent on the network [14] and optimizes the use of network resources while ensuring reliable service. [4] The transport layer breaks up messages into packets to ensure that no station monopolizes the channel while sending a long message (e.g., a file). The source transport layer attaches a transport header and trailer to each packet. (These differ from the session layer header which is attached to the entire message. The session layer header is treated as part of the data by the transport layer.) The header includes a start of packet indicator, transport source address and transport destination address, the number of packets in the message, the packet sequence number, and routing information. The trailer contains an end of packet indicator and, if desired, cyclic redundancy check (CRC) bits. [10]

The destination transport layer strips off the header and trailer and reassembles the message before passing it up to the session layer. The transport addresses typically contain a network number, a host number and a port number if hierarchical addressing is used or a unique address if flat addressing is used. These identify the communicating processes uniquely. The transport layer uses a name server to translate the name of a process requested by the session layer into the address where the process is located and can be accessed. The address can then be translated to the associated process name for session layer

use. [8] The transport layer sets up and takes down the connections between two processes.

## 6.1 Multiplexing

On a broadband LAN the transport layer may send packets for one connection over several channels to increase throughout. This is called downward multiplexing and is useful for large file transfers or to give one user higher priority. If the host has multiple ports with sporadic (e.g. interactive) traffic, the transport layer may multiplex several connections over the same channel. This is called upward multiplexing.

## 6.2 Flow Control

The transport layer is also responsible for end-to-end flow control - ensuring that a fast host does not overwhelm a slow host. Flow control is accomplished by allowing only a limited number of packets to be sent without receiving positive acknowledgement from the receiver that the packet was correctly received. The number of packets that can be unacknowledged at any time is called the window size. The window size is closely associated with the number of buffers available in the hosts. If the fast sender has 50 buffers for outgoing traffic but the slow receiver has only 5 buffers for incoming packets, the window size can be no more than 5; if more than 5 packets are sent the receiver will have to discard some packets because it has no memory to hold them. As the receiver's buffers are emptied it can signal the sender to allow more packets to be sent. Similarly, if the sender has

fewer buffers than the receiver, the smaller number must be used for the maximum window size. To see why consider what would happen if a packet was lost or damaged. The sender must retransmit that packet. Thus the sender must keep a copy of every outstanding packet. The number of packets that can be stored for possible retransmission is the number of buffers. Buffer management is an important part of the transport layer.

## 6.3 Error Checking

How does the transport layer know when a packet has been damaged? For example, the change of one bit can change a 5 to a 7. The data link layer has responsibility for error-checking but this may not catch enough errors for certain applications. The transport layer will then add its own error-checking on top of the error-checking of the data link layer. The procedure is identical to the one used in the data link layer and is explained in that discussion. If an error is detected at this level, a request for retransmission is sent to the sending host.

## 6.4 Gateways and Bridges

Although ISO has approved the Open Systems Interconnection - Connection Oriented Transport Protocol, it has not yet received industry-wide acceptance. Most LANs in the United States use either the Department of Defense's Transmission Control Protocol (TCP) or the Xerox Network System (XNS) protocol (which has become the de facto standard for Ethernet networks) or they develop their own proprietary

transport layer. [10] Thus, if you wish to connect two different LANs, you need a gateway which converts packets from one protocol to another or two half-gateways, one at each LAN to convert packets from the internal protocol to a standard packet for internetwork traffic and back. [8] If the two LAN's use identical protocols, a bridge or two half-bridges can be used. Since they need only account for flow control and congestion control between the networks, they can be much simpler. [10] Half-gateways and Half-bridges have the advantage that they can be incorporated into a host on a LAN instead of having a separate machine performing this task. Also, internetwork traffic is unaffected if a LAN changes its protocols; the LAN simply changes its half-gateway to match the new protocols and other LANs will not notice the change.

## 7. THE NETWORK LAYER

The network layer deals with network management and control functions. It is not an end-to-end protocol. Instead it controls the exchange of messages hop by hop along the network and defines the host-network interface. Concerns at this level include packet routing, congestion control, minimizing delay while maximizing throughtput, fault isolation and diagnosis and network statistics. This layer is less important in LANs than in long-haul networks because of the special characteristics of LANs.

For example, packet routing need not be considered on a bus or ring network where every station receives every packet. Only a network with a mesh topology need worry about routing decisions. Similarly, the access methods typically used on LAN's, CSMA/CD and token-passing have a built-in mechanism to avoid congestion (as will be shown later) so congestion control is not needed. The access method and physical medium used determine the network initiated delays and the maximum throughput.

The important functions performed at this layer on a LAN are transparent error detection and correction, maintaining network statistics, and allowing for network-wide maintenance. [10] The network layer can arrange for periodic loop backs to check for faulty devices and signal other stations when a device has failed. [4] If necessary, rerouting can be specified. Control messages can also be

sent between adjacent stations asking if the station is okay. If no response is received in a specified time, the neighboring station will announce to all stations that the station has failed. [1] Finally, statistics can be maintained on network use, actual delays, actual throughput, traffic patterns, types of transactions, etc., which can be used by the network manager to alter the services provided, plan for future expansion, detect possible problems, etc.

Those network layer functions that have been implemented on LANs are network specific and often implemented in VLSI chips. This also limits internetworking. A standard network layer for LANs will probably be developed only after several years of experimenting with various implementations.

# 8. THE DATA LINK LAYER

The task of the data link layer is to provide error-free transmission from one station to the next on the LAN in a manner which makes the physical medium transparent to the network layer. For LANs the main concerns of this layer are media access, error handling, frame handling and bit synchronization. Three incompatible standards have been defined by the IEEE 802 Committee dealing with this layer with several other proposals under consideration. The three standards differ in the access method used but share a common logical link control sublayer, so the network layer may be the same for all three standards.

## 8.1 Logical Link Control Sublayer

The logical link control sublayer specifies the interface between the data link layer and the network layer and the interface between this sublayer and the media access control sublayer. Two types of services are available to the network layer, a connectionless, datagram style service in which there is no acknowledgement (at this layer) of the receipt of a correct frame and an optional connection-oriented service in which frames are acknowledged. [10] (A frame is a packet to which this layer has added its own header and trailer.) All stations are required to support connectionless service, but not the connection-oriented service. When connectionless service is utilized higher-level protocols must ensure that packets are delivered without error. Since a single station may have several processes running

concurrently servicing or communicating with other processes at other stations, the protocol must identify to which exchange a message belongs. For this purpose unique service access point (SAP) addresses are used. This address is a port to a higher level protocol. Both the source and destination SAP are specified. The interface also uses the HDLC control commands/responses unnumbered information (for connectionless service) and receive ready, reject and receive not ready (for connection-oriented service). [10] This information possibly including supervisory information is added as a header to the packet received from the network layer at the source station and removed by the sublayer at the destination station. [3]

## 8.2 CSMA/CD Bus

The first media access control sublayer defined is virtually identical to the Ethernet standard previously agreed to by the Xerox, Digital Equipment and Intel Corporations. The changes made were accepted by the corporations for their Ethernet products. This standard uses the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) access protocol on a bus topology. CSMA/CD is a contention protocol in which all stations contend for the right to transmit when the channel is free. The protocol is as follows. A station having a frame to send monitors the carrier sense signal on the channel. If the channel is free, it starts transmitting its frame and listens for a collision for at least the round-trip propagation delay plus the collision detection hardware delay. This determines the minimum frame size on a certain implementation. [3] (For example, the minimum frame

size on a 10 Mbps (megabit per second) baseband system is 512 bits. [10]) If no collision occurs the station has acquired the channel.

Stations that find the channel busy when they listen must wait until the channel is free. It two stations try to acquire the channel during the same frame slot (called the collision window), a collision will occur - garbling both messages. When a collision is detected the collision-detect signal is turned on and the MAC sublayer starts sending a "jam" signal for an implementation-specific time to ensure that all stations will notice the collision. All stations involved in the collision must now wait a pseudo-random number of frame slots before attempting to retransmit.

If the network is very busy, many stations may attempt to transmit during one slot. If the competing stations only wait an average of 2 or 3 slots after a collision, repeated collisions will occur and the network will eventually come to a standstill; no useful information will pass through the network. To ensure that this does not occur while still allowing good throughput at light load, an algorithm called binary exponential backoff is used. When the first collision occurs each contending station chooses its pseudo-random number from 0 or 1. If successive collisions occur, the station is required to double the number of choices from which to pick the wait-time for each new collision. That is, the number of slots that a station must wait is an integer, w such that $0 \leq w < 2^n$, where n is the number of successive collisions the station has encountered. Since the maximum number of

stations allowed is $2^{10}$ = 1024, n is not allowed to exceed 10. The standard specifies that a station should abandon its retransmission attempts after 16 tries and signal an error condition since clearly either the network or a station has malfunctioned. [10]

The frame created and sent by this layer is shown in Figure 4.

| | preamble | SD | DA | SA | L | data | pad | FCS |
|---|---|---|---|---|---|---|---|---|
| # octets | 7 | 1 | 2or6 | 2or6 | 2 | | | 4 |

FIGURE 7

The preamble is 28 01 bit pairs used to allow the stations to become synchronized and to allow the physical link signaling circuitry to reach its steady state. [10] The start of frame delimiter (SD) is 01010111 and signals the start of a valid frame. The source and destination addresses (SA and DA) may be local or global (unique) addresses. All stations must use the same number of octets for addressing - either 2 octets or 6 octets. Global addresses are assigned by Xerox Corporation to ensure that there are no duplicate addresses, so internetwork packets are sure to get to the correct station. Since all frames in a particular ethernet LAN are the same size, the source station may have to add padding (pad) to fill out the frame. [3] The length field (L) indicates the number of octets of useful data in the frame so the destination station can strip off the padding. Since all frames are the same size, there is no need for an

end delimiter to mark the end of a frame. The frame check sequence (FCS) is a 32-bit field used to check for errors during transmission. The generator polynomial used is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. [3]

## 8.3 Token Bus

The second media access control sublayer defined by the standard is a token bus. The bus topology described above is used in this sublayer, also. The difference is that stations are dynamically ordered to form a logical (as opposed to a physical) ring. [15] Each station maintains the addresses of its predecessor and successor stations. The logical ring need not have any relation to the physical positions of the stations on the bus. This access method is not a contention protocol. A station may transmit only when it has captured a special bit pattern called the token. While a station has the token it may transmit packets up to the maximum number allowed by a certain implementation. It then regenerates the token and sends it to its successor. If the next station on the logical ring has nothing to send, it forwards the token to its successor. (The token includes source and destination addresses.)

Four priority classes of stations may be set up on a token bus LAN. Each station is required to time the rotation of the token around the ring. Under heavy load the token will take a relatively long time to return to a station. Low priority stations are given a short target time. If the token rotation time is more than the target time, the

station may not capture the token. Thus, at light load all stations may transmit, but at heavy load only high priority stations may transmit.

Token access is a much more complex protocol than CSMA/CD, because the network will fail if the logical ring is broken. The protocol must handle such tasks as initiating the token passing sequence, detecting and regenerating lost tokens, detecting and eliminating multiple tokens, and adding and deleting stations. To detect lost tokens a station is required to set a timer and listen after forwarding the token. [16] If the station hears a frame being sent prior to the timer timing out, the token pass must have succeeded. If the timer goes off and no transmission is detected, either the token is lost or damaged or the successor station has failed. The station's first response is to resend the token to its successor. If the previous token was damaged by a noise burst, this one will probably be successful and the successor will capture the token or forward it. If the timer times out a second time, the assumption is that the successor has malfunctioned. The station now sends a control frame asking who is the successor of the station's successor. (Recall that each station maintains the address of its predecessor.) If this station responds, the logical ring is reconfigured (the original station changes its successor address and the second station changes its predecessor address) and the token is sent to the new successor. If this fails twice the logical ring must be re-initialized as indicated below. [10]

Initializing the logical ring and adding stations to the ring use the same contention algorithm. During initialization one station (triggered by a timer) sends out a request for a successor over a range of stations. This process is repeated until the logical ring is closed. After the logical ring is established, each station is required to send such a frame periodically to see if new stations want to be added to the ring. [17] After every media access control (MAC) frame, the sender must listen a specified time period for a response. If no station starts transmitting during this response window, no stations within the specified range of addresses wish to be added and the logical ring remains unchanged. Contention occurs when several stations wish to be added and a collision occurs. When the stations hear the collision they choose a pseudo-random integer between 0 and 3 and must wait that many slots before again responding. This process is continued until only one station responds and is added to the ring and given the token. [10]

There are a number of differences in the frames for the token bus and the frames for the ethernet bus. The token bus frames are all the same length. Thus, padding and a length count are unnecessary. However, a special bit pattern is needed to indicate the end of a frame. Other differences are that the preamble may be one or more octets, not the required seven in the ethernet and one octet is used to indicate the type of frame (e.g., token, control, or data.) [17]

## 8.4 Token Ring

The last MAC sublayer in the standard is the token ring. In this instance access to the physical medium is also controlled by the use of a token, but the ring is now a physical ring. That is, the stations on the network are connected to a channel formed into a ring and stations are ordered by their physical location on the ring. While the two bus LANs use a passive interface in which a station simply listens to messages on the channel and copies messages sent to it or puts messages onto the channel, the token ring uses an active interface. While special hardware removes messages from the bus, it is the responsibility of the sending station to remove a message after it transverses the ring. Also, each station examines and then repeats, changes, or removes each bit it receives.

The token ring token is a 24-bit sequence which can be changed from a token to a connector by changing the last bit. To allow higher priority messages greater access to the channel when many stations have messages to send, the token can be marked as having one of four priorities. Only stations with that priority or higher may capture the token (by changing the last bit) to send their messages. [10]

The token ring frame is very similar to the token bus frame as you can see in Figure 8. The differences are in the start frame and end frame delimiters, the access control field (AC) and the lack of a preamble.

| SD | AC | DA | SA | Data | FCS | ED |
|---|---|---|---|---|---|---|
| 2 | 1 | 2or6 | 2or6 | 0-4099 | 4 | 2 |

FIGURE 8

The first three octets comprise the token when a particular pattern appears in the access control field. Two bits in the end frame delimiter are used by the receiving station to acknowledge messages sent to it. One bit is turned on (changed from a 0 to a 1) to indicate that the station recognized its address and attempted to copy the message. The second bit is turned on if the message was copied successfully and the check sequence balanced. [10]

## 8.5 Error Handling

No matter which access method is used, this layer must be able to detect and handle most errors that occur. Errors can be caused by noise on the line (electrical surges), transmitter or receiver failures, station failure or malfunctions, insufficient buffers, repeater failures or failure of the channel. All three access methods use a cyclic redundancy check sequence with the generator polynomial $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$. The basic concept behind the CRC is as follows. All the bits in the frame prior to the frame check sequence (FCS) are treated as the coefficients of a polynomial. To these bits are added "0" bits. The number of "0" bits added is equal to the degree of the generator polynomial (32 in this case). This

"polynomial" is then divided by the generator polynomial and the remainder is subtracted from the "polynomial" modulo 2. All subtraction done as part of the division process is also done modulo 2. [8]

Subtraction modulo 2 (without borrowing) is equivalent to exclusive or in which if two bits are identical the answer is 0 while if they are different the answer is 1. For example, in modulo 2 we have:

```
  0      1      1      0      1011
 -0     -0     -1     -1     -1101
  0      1      0      1      0110
```

The sending station divides its message by the generator polynomial using binary division and then subtracts the remainder from the original message (to which 32 zeros have already been appended). Thus, the frame sent (less the end frame delimiter, if it has one) if treated as a polynomial must be divisible by the generator polynomial. For example, in the figure below we see that the original number, 3500, is not divisible by 101. However, after subtracting the remainder the new number is divisible by 101.

```
      34                3500                 34
101)3500          -  66          101)3434
   303               3434             303
   470                                404
   404                                404
    66                                  0
```

FIGURE 9

The same concept is used with the frame check sequence which is the remainder which was subtracted from the original message. The

receiving station simply divides the message with its FCS by the generating polynomial. If the remainder is not zero, the frame has been damaged.

This division has been implemented in hardware using a shift register and can be performed very rapidly. The CRC specified by the Project 802 standards will catch all burst errors of length less than or equal to 32 bits, all errors affecting an odd number of bits and 99.9999999 + % of all burst errors of length greater than 32 bits. In other words, very few errors go undetected. When an error is detected the frame is simply retransmitted.

# 9. THE PHYSICAL LAYER

The transmission of 0's and 1's over the communication channel is the concern of the physical layer. It is at this layer that the actual communication occurs. At all other layers the communication is virtual and is accomplished via headers and trailers. Issues at this layer include the interface between the physical and data link layers, signal levels, modulation technique used, choice of baseband or broadband implementation, electrical and mechanical specifications for the physical channel and the connectors, signal generators and receivers and baud rate. [8]

The Project 802 standards specify a family of standards for the physical layer for the three types of LANs described in the Data Link Layer section. The CSMA/CD bus standard has a baseband and a broadband version. The difference between baseband and broadband is that in a baseband network only one signal may utilize the communications medium at any instant while in a broadband network frequency division multiplexing (FDM) is used to divide the communication medium into frequency bands or channels and different signals may be on each channel. [12] Special modems are used to send and receive at different frequencies. [19]

## 9.1 CSMA/CD Bus

The baseband version uses 50 ohm (50Ω) shielded coaxial cable as the transmission medium and Manchester encoding and operates at 10

megabits per second (10 Mbps). [10] Manchester encoding is a technique in which both a data signal and the clock signal are sent in the same bit stream. This is accomplished by having two voltage readings per bit sent. The bit period or bit cell is 100 nanoseconds (100 ns). The signal sent during the first 50 ns is the bit value. The second 50 ns the opposite value is sent. This is to ensure that the clocks of the two stations remain synchronized. If the receiver samples the signal and hears two 1 bits or two 0 bits in the same bit cell, it must be out of synchronization and must resynchronize. -1.025 volts to represent a 0 and -2.05 volts to represent a 1 have been proposed. [10] The standard specifies maximum tolerances for each specification. This encoding allows for synchronous transmission and higher data rates than are possible with asynchronous transmission where a start bit and one or more stop bits are sent with each byte (octet) of data. With asynchronous transmission the two clocks need not be synchronized.

Stations are connected to the bus transmission medium using passive taps which do not cut or break the cable. These taps should occur only at markings on the cable that occur every 2.5 meters to ensure that there is no significant in-phase addition of signal reflections. A maximum of 100 taps is allowed per segment. Each segment may be up to 500 meters. Segments are connected by repeaters which regenerate the signal to form a branching nonrooted tree topology. The cable connecting the station to its transceiver that sends and receives signals on the bus is a 78Ω twisted pair cable with a male and female 15-pin D series connector at each end. This cable

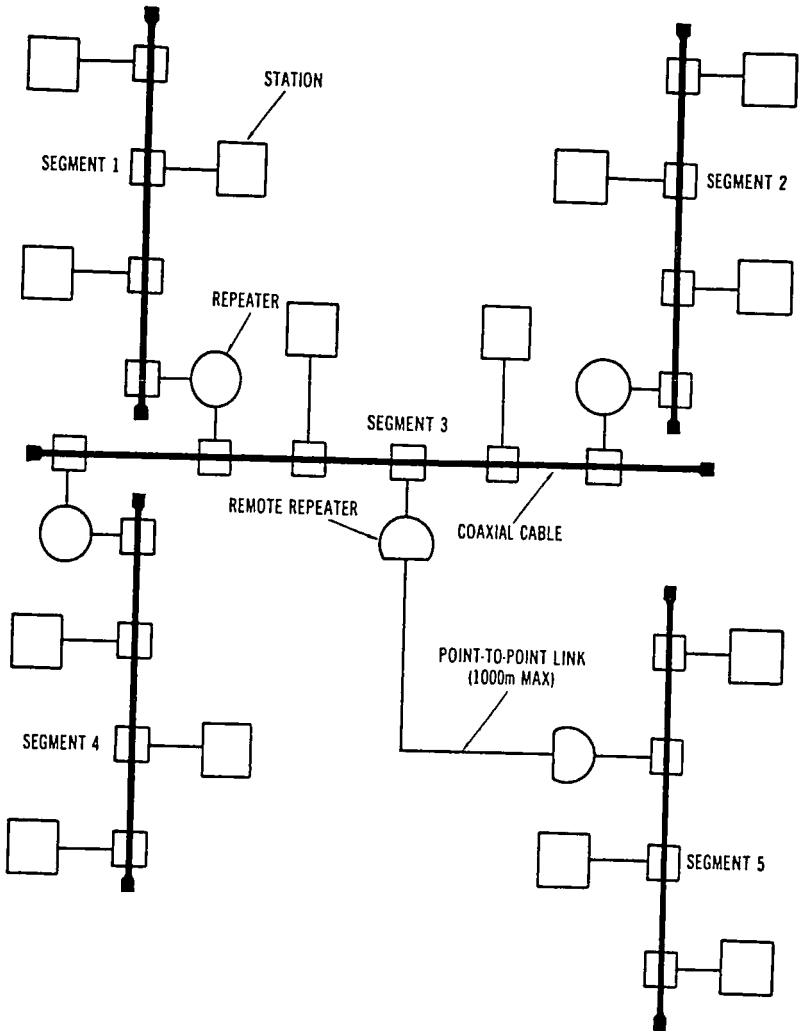



FIGURE 10

may not exceed 50 meters. Data can be transmitted to and from the transceiver at data rates of 1, 4, 5, 10, or 20 Mbps. A maximum of 1024 stations can be on the LAN and no two stations are allowed to be separated by more than two repeaters. A point-to-point link of up to 1000 meters is allowed to connect two widely separated segments. Repeaters are needed at each end of the link. Terminators are used at the ends of each segment to avoid signal reflection. [10]

The broadband version will use 75Ω coaxial cable. Special modems will be needed by stations to send and receive at different frequencies. Both 2 Mbps and 10 Mbps data rates have been proposed. The signaling method to use has not yet be determined. [10]

## 9.2 Token Bus

The token bus has two baseband versions and a broadband version. All three use 75Ω coaxial cable for the bus. The slower baseband version uses phase-continuous frequency-shift keying (FSK) and differential Manchester encoding to achieve a data rate of 1 Mbps. Bits are modulated using phase-continous FSK by smoothly varying the signal frequencies between 3.75 megahertz (MHz) and 6.25 MHz every 100 nanoseconds. [10] As noted before, Manchester encoding sends the clock signal with each data signal. With differential Manchester encoding, the frequency shift represents the change from the previous data signal. In this way multiple bits can be sent with each signal. Very short cables ($\leq$ 35 cm) connect stations to the bus. [10]

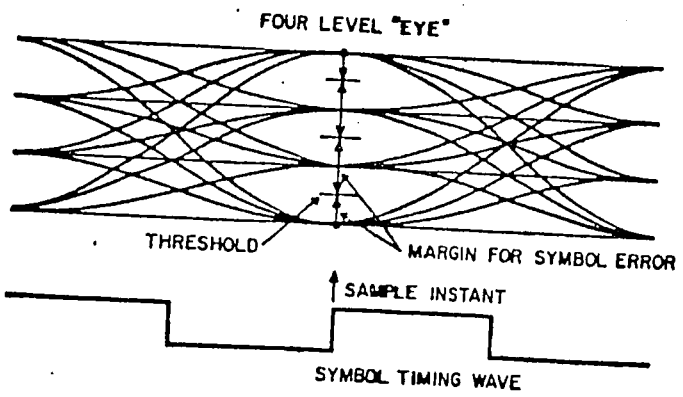



FIGURE 11

The second baseband version uses phase-coherent FSK to send 1 bit every 100 ns. With phase-coherent FSK frequencies are shifted abruptly at a zero crossing of the carrier. Data rates of 5 Mbps or 10 Mbps are achieved by using signaling frequencies of 5 MHz and 10 MHz (5 Mbps) or 10 Mhz and 20 MHz (10 Mbps). [10]

The broadband version uses a three-level duobinary amplitude modulation/phase-shift keyed (AM/PSK) modulation scheme. [10] Amplitude modulation refers to changing the amplitude of the sine wave to represent different bits. Phase-shift keying refers to an abrupt change in the phase of the sine wave at the boundary between data signals. Duobinary refers to a technique used to shape the signal via filters to reduce intersymbol interference and thus reduce the frequency spectrum required. [20] By sampling the signal at the widest part of the "eye" the probability of error is reduced. [20]

## 9.3 Token Ring

The physical layer for the token ring is still under consideration. Two baseband versions have been proposed. The cheaper version uses 150Ω shielded twisted pair as its transmission medium. It operates at 1 Mbps or 4 Mbps using differential Manchester encoding. The second version uses coaxial cable and differential Manchester encoding to achieve data rates of 4 Mbps, 20 Mbps or 40 Mbps.

## 9.4 Future Trends

In addition to this family of standards we can expect other versions of these LANs in the future that use different transmission media. For example, fiber optics has many advantages over cable. High bandwidth is possible with fiber optics because it is capable of billions of cycles per second. It is unaffected by electric and magnetic fields and can easily be tested for imperfections. Also, it is much more secure than cable since no radiation escapes that can be secretly monitored. [21] Or infrared light may prove to be an inexpensive alternative for private branch exchange (PBX) LAN's where use of a broadcast technique is plausible. [22]

# 10. CONCLUSION

In this thesis distributed processing has been defined as the use of a transmission medium to share computing responsibilities among several computers in such a way that the computers are "equals" and such that the distribution of responsibility is transparent to the user. The need for standardization and open systems was demonstrated by showing the unlikelihood of interconnection of many incompatible LANs and the desirability of such interconnection. The ISO-OSI Reference Model and IEEE Project 802 Committee standards were presented as the probable choice for such standardization.

The seven layers of the OSI Reference Model were presented in a top-down fashion. The application layer which deals with application-oriented functions to be included in a network library was described. It was noted that the presentation layer provides often-used services, while the session layer manages the dialog between two processes. We have shown that the transport layer manages end-to-end packet exchanges and that the network layer deals with network management. The data link layer which provides media access and error-free transmission and the physical layer which transmits bits over the physical channel were described from the perspective of the IEEE Project 802 Committee standards.

The thesis gives a detailed description of the IEEE Project 802 standards that have been published to date and predicts probable

choices for some aspects not yet standardized. Three incompatible standards were presented for a CSMA/CD bus (Ethernet), a token bus and a token ring. All three share a common logical link control sublayer. Two versions of the CSMA/CD bus were presented - a baseband version and a broadband version, while two baseband versions have been proposed for the token ring.

REFERENCES


1. Kahne, S., I. Lefkowitz and C. Rose: "Automatic Control by Distributed Intelligence", Scientific American, June, 1979.

2. Stankovic, J. A.: "A Perspective on Distributed Computer Systems", IEEE Transactions on Computers, December, 1984, pp. 1102ff.

3. Harrison, T. J.: "IEEE Project 802: Local Area Network Standard", March, 1982, Status Report.

4. Green, P. E., editor: Computer Network Architectures and Protocols, c 1982, Plenum Press.

5. Haugdahl, J. S.: "Local Area Networks for the IBM PC", BYTE, December, 1984, pp. 147ff.

6. Daney, C.: "Protocols", Computers and Electronics, May, 1984, p. 63, pp. 103ff.

7. Mier, E. E.: "The Evolution of a Standard Ethernet", BYTE, December, 1984, pp. 131ff.

8. Tanenbaum, A. S.: Computer Networks, c 1981, Prentice-Hall.

9. McKenzie, A. and H. Forsdick: "File Transfer Protocol Specification", July, 1979, BBN Report Number 4051.

10. Bartee, T. C., editor: Data Communications, Networks, and Systems, c 1985, Howard W. Sams and Co.

11. Nicolai, C.: "Encryption Decyphered", Computers and Electronics, June, 1984, pp. 64ff.

12. Daney, C.: "Mainframe Communications Terminals", Computers and Electronics, May, 1984, pp. 102-3.

13. Gabel, D.: "The Mainframe Connection", Personal Computing, September, 1984, pp. 157ff.

14. Gugliotti, J. A. and E. B. Weitz: "Getting Mainframe Data to Micros", Computers and Electronics, May, 1984, pp. 61ff.

15. Fine, M.: "Demand Assignment Multiple Access Schemes in Broadcast Bus Local Area Networks", IEEE Transactions on Computers, December, 1984, pp. 1130ff.

16. Byers, T. J.: "Choosing a Local Area Network", Computers and Electronics, June, 1984, pp. 72ff.

17. Phinney, T. L. and G. D. Jelatis: "Error Handling in the IEEE 802 Token-Passing Bus LAN", IEEE Journal on Selected Areas of Communications, November, 1983, pp. 785ff.

18. Kong, I. and B. Lough: "Local Area Network - A broadband Implementation".

19. Byers, T. J.: "New Low Cost Modems", Computers and Electronics, May, 1984, pp. 49ff.

20. Green, P. E. and R. W. Lucky, editors: Computer Communications, c 1975, IEEE Press

21. Shuford, R. S.: "An Introduction to Fiber Optics, Part I", BYTE, December, 1984, pp. 121ff.

22. Byers, T. J.: "Electronic Ties That Bind", Computers and Electronics, March, 1984, pp. 68ff.

VITA

Dennis C. Ebersole was born on April 3, 1949 in Lancaster, Pennsylvania to Galen and Melva Ebersole. He attended Bucknell University from 1967 through 1971 as a National Merit Scholar - receiving a B.S. in mathematics in 1971. In 1977, he received his M.S. in mathematics from Lehigh University.

For the past fourteen years he has been a member of the mathematics department at Northampton County Area Community College where he is currently an associate professor and will be promoted to professor in the fall, 1985. He was the recipient of an Excellence in Teaching Award for 1985.

Dennis is the author of a textbook, Shop Mathematics published by Prentice-Hall, Inc. In addition to being on College committees, including chairing the College Senate and the Instructional Resources Committee, he has also been active in professional organizations. He is a Pennsylvania delegate and chairman of the Student Learning Problems committee in the American Mathematical Association of Two Year Colleges, was president of the Pennsylvania State Mathematics Association of Two-Year Colleges for two terms and is currently the vice president and editor of its Two Year College Directory. He is a member of the executive board and the program committee of the Pennsylvania Council of Teachers of Mathematics.